# Happy last lecture!

- Quiz 12, Friday, Nov 20th 6am until Nov 22nd 11:59pm (midnight)
  - Practical advice
  - This quiz will have 10 questions (15min)
  - If you get 5 questions correctly, you get a full grade on it
- Touch-point 3: deliverables due Nov 22nd, live-event Mon, Nov 23rd
  - Single-slide presentation outlining progress highlights and current challenges
  - Three-minute pre-recorded presentation with your progress and current challenges
- Project final report due Dec 7th 11:59pm (midnight)
  - GitHub page with all of the results you have achieved utilizing both unsupervised learning and supervised learning
  - Final seven-minute long pre-recorded presentation
- CIOS participation (> 80% of the class by 12/11) → <u>1 bonus point for everyone</u>

# Grading schedule

- Assignment 4 grades → 12/4 by 11:59 pm
- Project final report → 12/9 by 11:59 pm

All regrade requests should be in by 11:59pm EST on 12/10

# Course objectives

- Introduce you to the machine learning **workflow**
- Develop deep understanding of major machine learning **algorithms**
- Learn how to **apply tools for real-data** analysis problems
- Create **effective visualizations** for data modeling
- Improve your written and verbal **communication skills**
- Experience **teamwork in a remote environment**
- Motivate you to do **research in data science** and machine learning

CS4641B Machine Learning

# Lecture 24: Practical advice

Rodrigo Borela ▸ rborelav@gatech.edu

Slides adapted from Vivek Srikumar, Andrew Ng and Mahdi Roozbahani

# Machine learning as a discipline

Study of algorithms that
- improve their **performance** P
- at some **task** T
- with **experience** E

well-defined learning task: <P, T, E>

— Tom Mitchell

# Unsupervised learning

- Probability and statistics and information theory
  - Covariance and correlation matrices
  - Entropy and mutual information
- Clustering
  - K-Means
  - DBSCAN
  - Probabilistic (using GMM)
  - Hierarchical
  - Clustering evaluation
- Probability density estimation
  - Parametric (exponential, Bernoulli, Gaussian)
  - Non-parametric (histograms, kernel density estimation)
- Dimensionality reduction
  - Feature selection
  - Principal component analysis

# Supervised learning

- Regression
  - Linear regression
  - Regularized linear regression
  - Neural Networks
  - Regression trees*
  - Convolutional neural networks
- Classification
  - Logistic regression
  - Bayes classifiers
  - Decision tree and random forest
  - Support vector machines
  - Kernel SVM
  - Neural networks
  - Convolutional neural networks

# Machine learning in practice

Machine learning is the process of **turning data into actionable knowledge** for **task support** and **decision making**.

Define your problem → Model your data → Evaluate → Deploy

# Outline

- Model diagnostics
- Error analysis
- Additional advice

# Outline

- Model diagnostics
- Error analysis
- Additional advice

# Debugging machine learning

- Suppose you train an SVM or a logistic regression classifier for spam detection
- You **obviously** follow best practices for finding hyperparameters (such as cross-validation) … and make sure there are no bugs in the code
- Your classifier is only 75% accurate

What can you do to improve it?

# Different ways to improve your model

- More training data

- Features
  1. Use more features
  2. Use fewer features
  3. Use other features

- Better training
  1. Run for more iterations
  2. Use a different algorithm
  3. Use a different classifier
  4. Play with regularization

## Tedious!

- And prone to errors, dependence on luck
- Let us try to make this process more methodical

# First step: diagnose your model

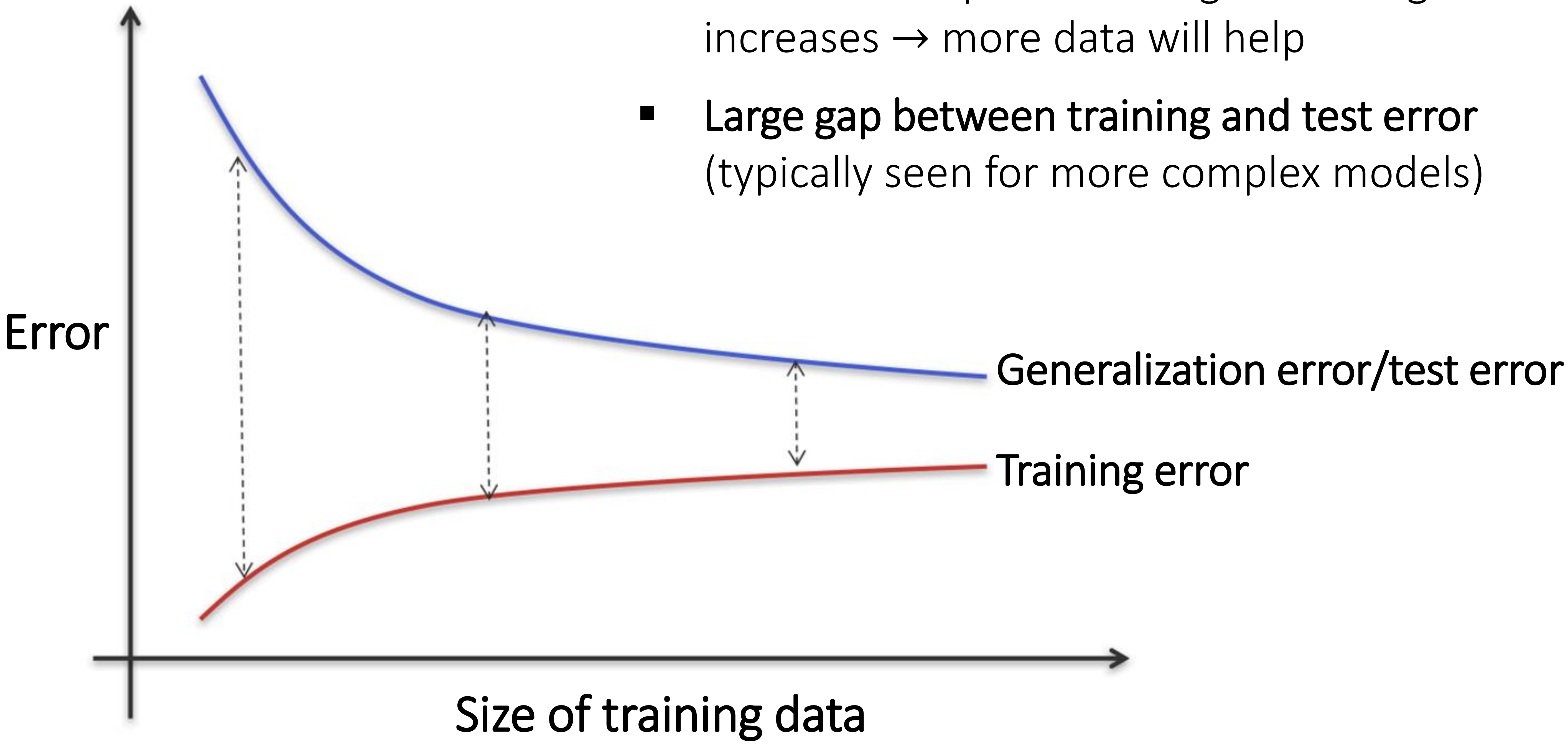
- Some possible problems:
  - Overfitting (high variance)
  - Underfitting (high bias)
  - Your learning does not converge
  - Are you measuring the right thing?

# Overfitting vs. underfitting

- Overfitting: the training accuracy is much higher than the test accuracy
  - The model explains the training set very well, but poor generalization
- Underfitting: both accuracies are unacceptably low
  - The model can not represent the concept well enough

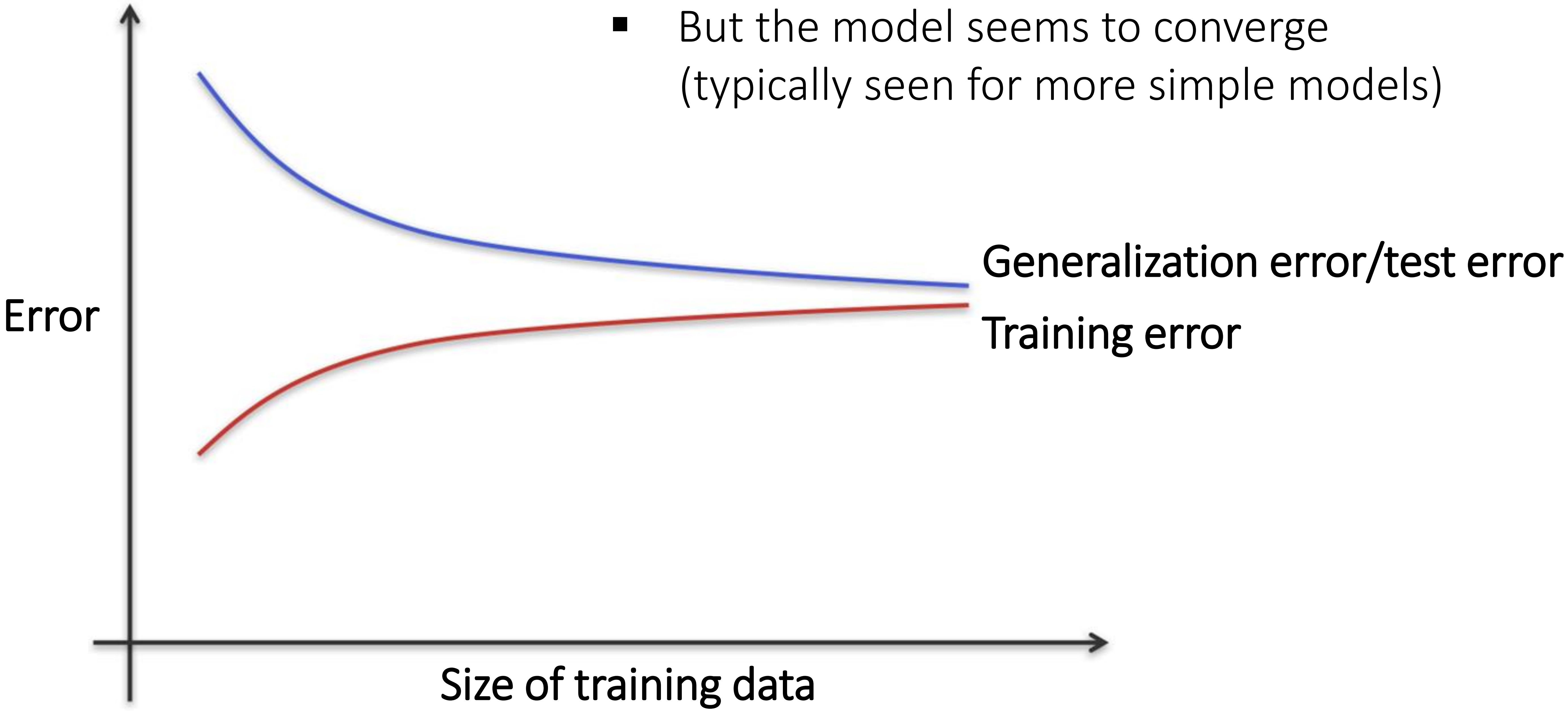
# Overfitting (high variance)

- Test error keeps decreasing as training set increases → more data will help
- **Large gap between training and test error** (typically seen for more complex models)



# Underfitting (high bias)

- **Both the train and test error are unacceptable**
- But the model seems to converge (typically seen for more simple models)



# Different ways to improve your model

- More training data → Helps with overfitting

- Features
  1. Use more features → Helps with underfitting
  2. Use fewer features → Helps with overfitting
  3. Use other features → Could help with overfitting and underfitting

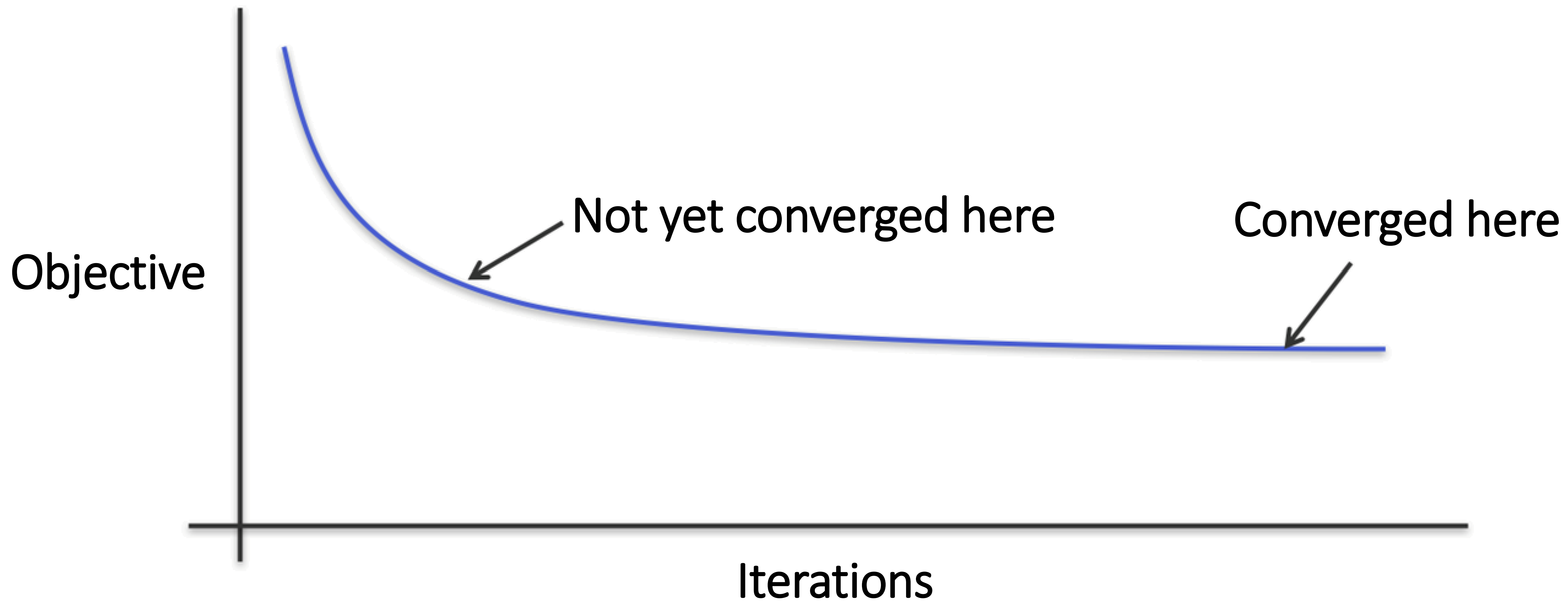
- Better training
  1. Run for more iterations
  2. Use a different algorithm
  3. Use a different classifier
  4. Play with regularization → Could help with overfitting and underfitting

# Diagnostics

- ✓ Overfitting (high variance)
- ✓ Underfitting (high bias)
- Your learning does not converge
- Are you measuring the right thing?

# Has the model converged?

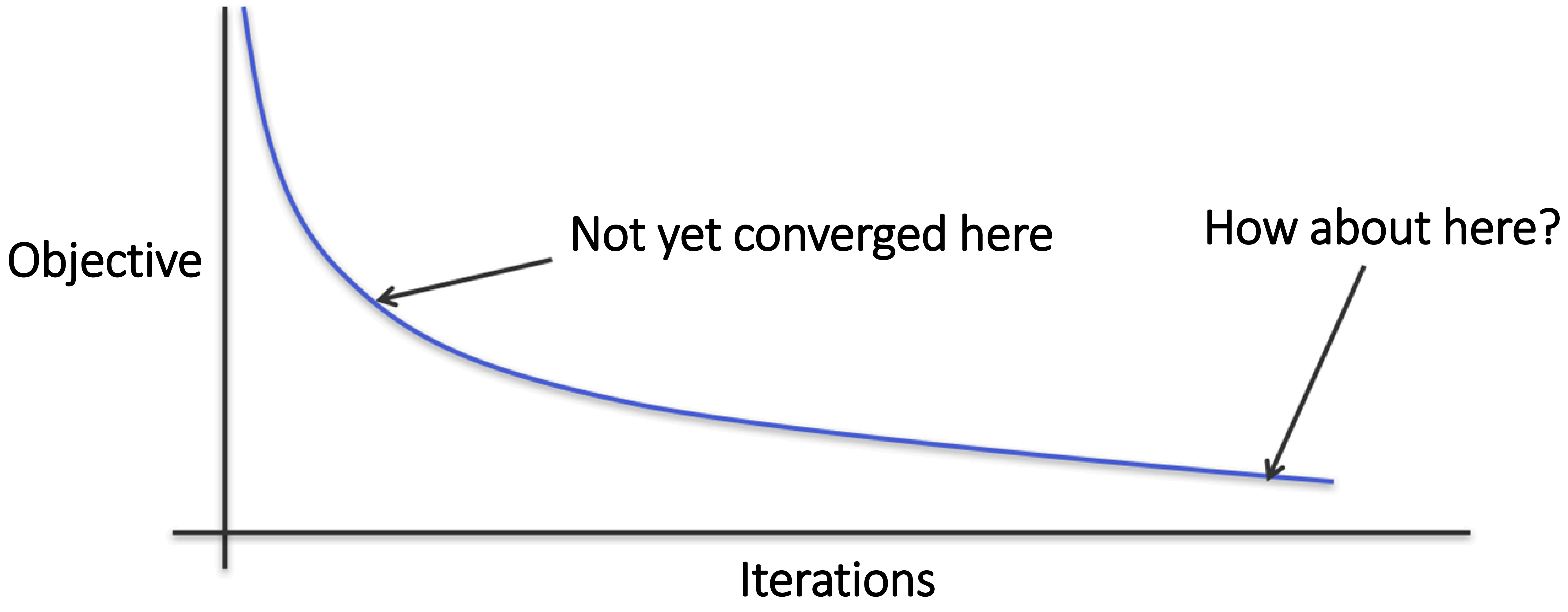
- If learning is framed as an optimization problem, track the objective



# Has the model converged?

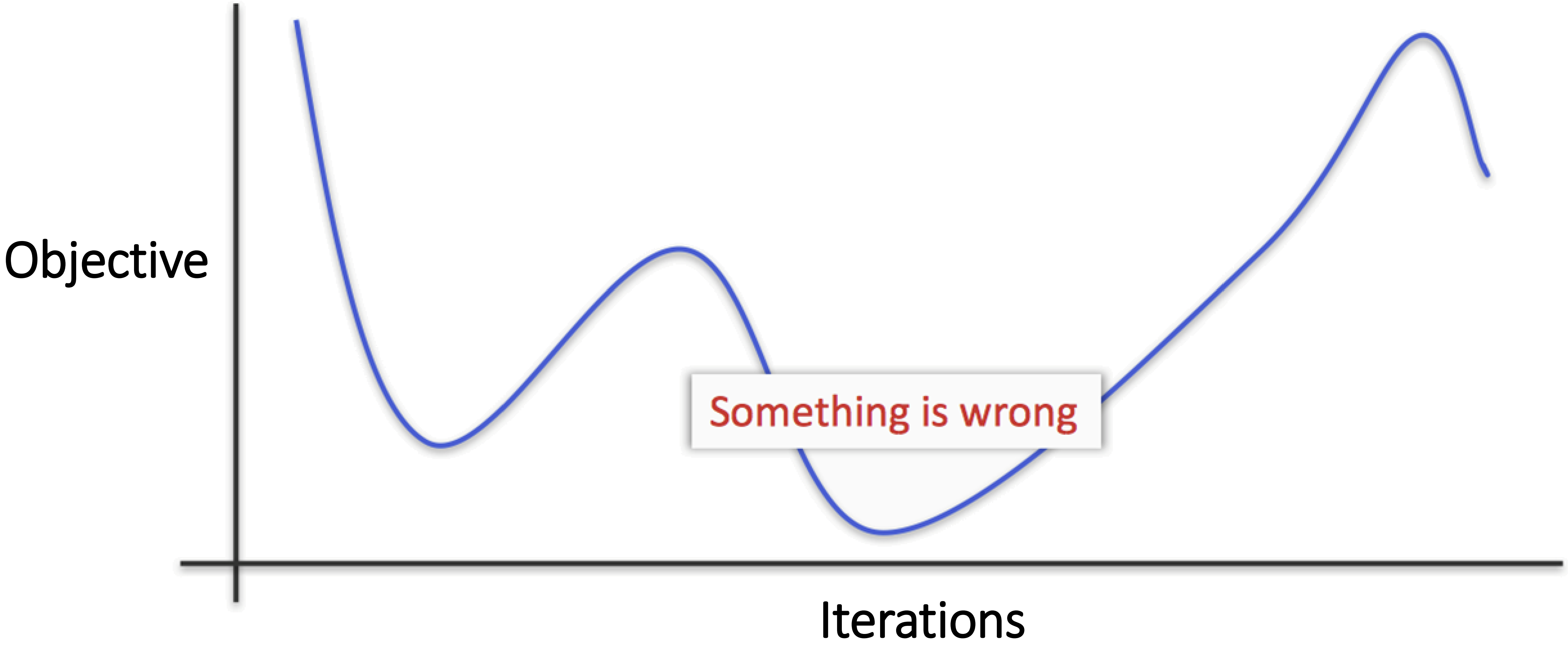
- If learning is framed as an optimization problem, track the objective

<u>Not always easy to decide</u>



# Has the model converged?

- If learning is framed as an optimization problem, track the objective
- **Helps to debug:** if we are doing gradient descent on a **convex** function the objective can't increase
- Important caveat: for SGD, the objective will slightly increase occasionally



# Different ways to improve your model

- More training data → **Helps with overfitting**

- Features
  1. Use more features → **Helps with underfitting**
  2. Use fewer features → **Helps with overfitting**
  3. Use other features → **Could help with overfitting and underfitting**

- Better training
  1. Run for more iterations
  2. Use a different algorithm → **Track the objective for convergence**
  3. Use a different classifier
  4. Play with regularization → **Could help with overfitting and underfitting**

# Diagnostics

- ✓ Overfitting (high variance)
- ✓ Underfitting (high bias)
- ✓ Your learning does not converge
- ▪ Are you measuring the right thing?

# What to measure?

- Accuracy of prediction is the most common measurement

- But if your dataset is unbalanced, accuracy may be misleading
  - 1,000 positive examples, 1 negative example
  - A classifier that always predicts positive will get 99.9% accuracy. Has it really learned anything?

- Unbalanced labels → measure label specific precision, recall and F-measure
  - Precision for a label: among examples that are predicted with label, what fraction are correct
  - Recall for a label: among the examples with given ground truth label, what fraction are correct
  - F-measure: harmonic mean of precision and recall

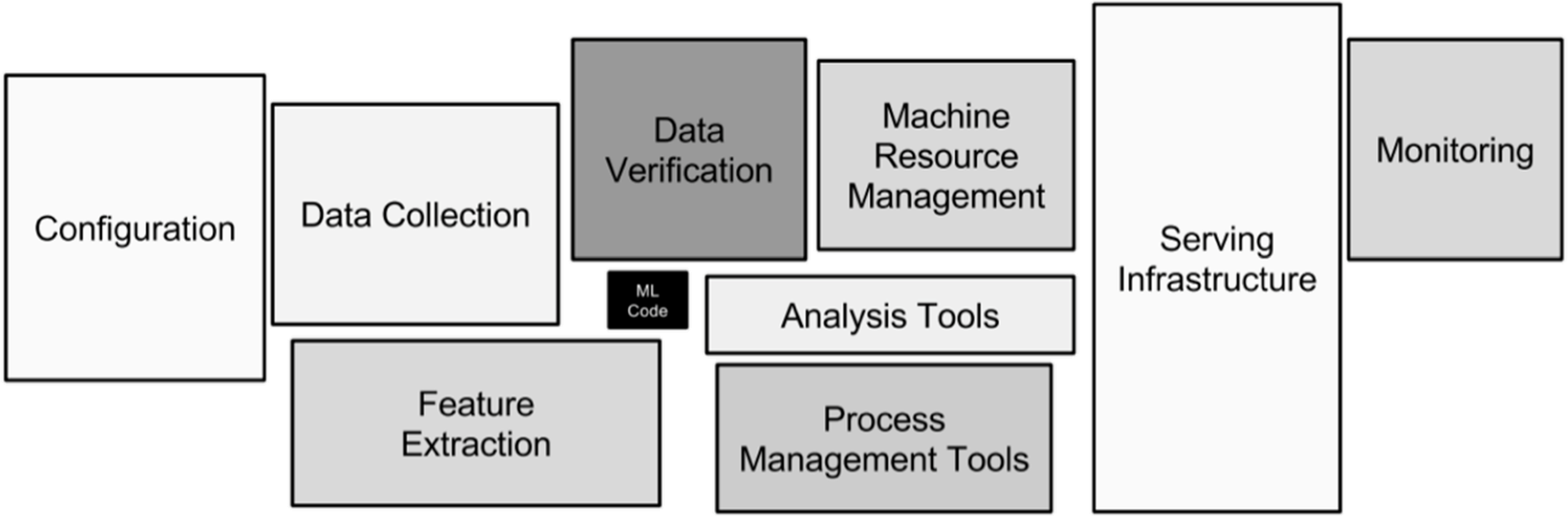
# Diagnostics

- ✓ Overfitting (high variance)
- ✓ Underfitting (high bias)
- ✓ Your learning does not converge
- ✓ Are you measuring the right thing?

# Outline

- Model diagnostics
- Error analysis
- Additional advice

# Machine learning in context



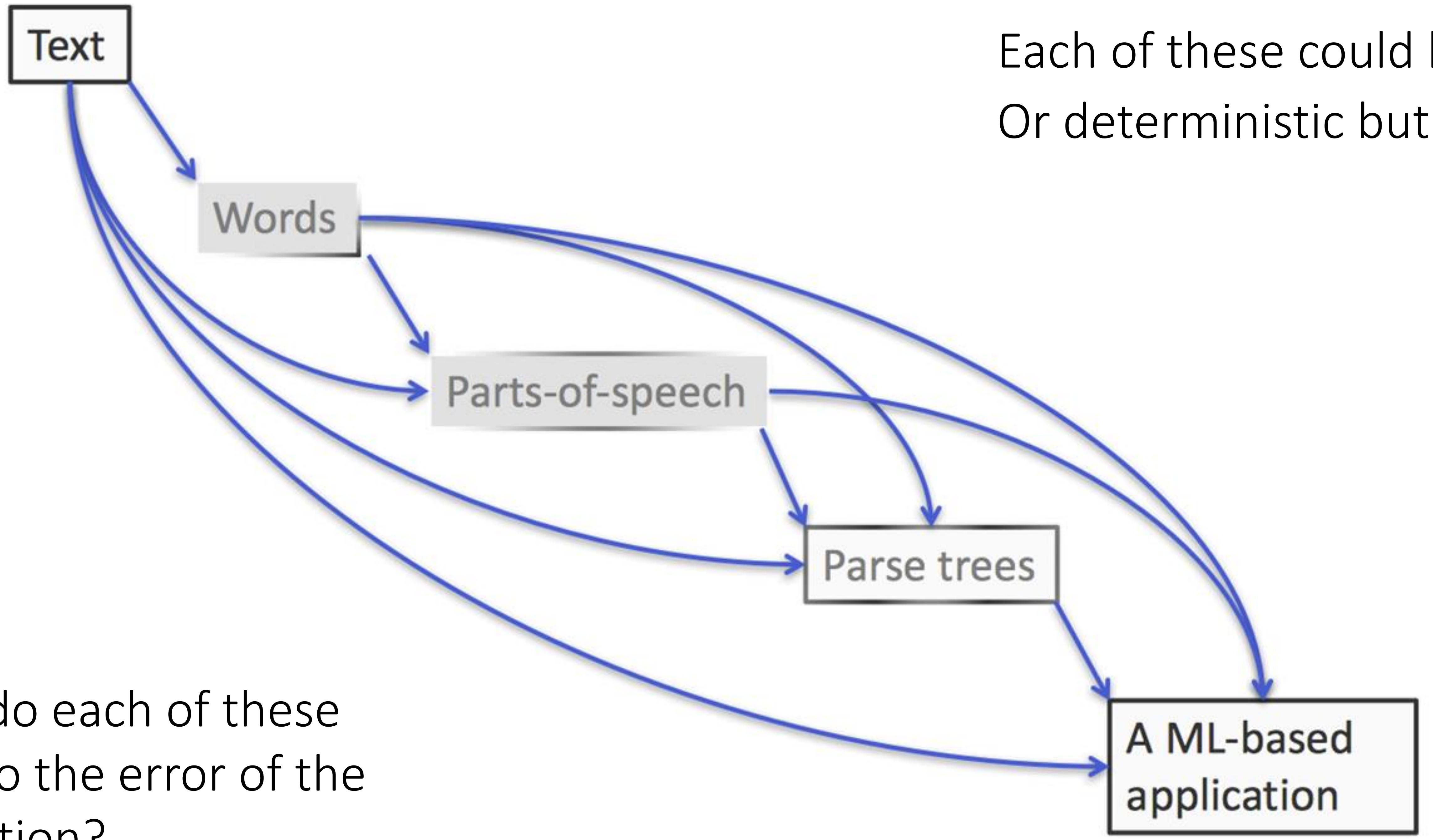
# Error analysis

- Generally machine learning plays a small role in a larger application
  - Pre-processing
  - Feature extraction (possibly by other ML based methods)
  - Data transformations (possibly by other ML based methods)

- How much do each of these contribute to the error?

- **Error analysis** tries to explain why a system is not performing perfectly

# Example: text processing system



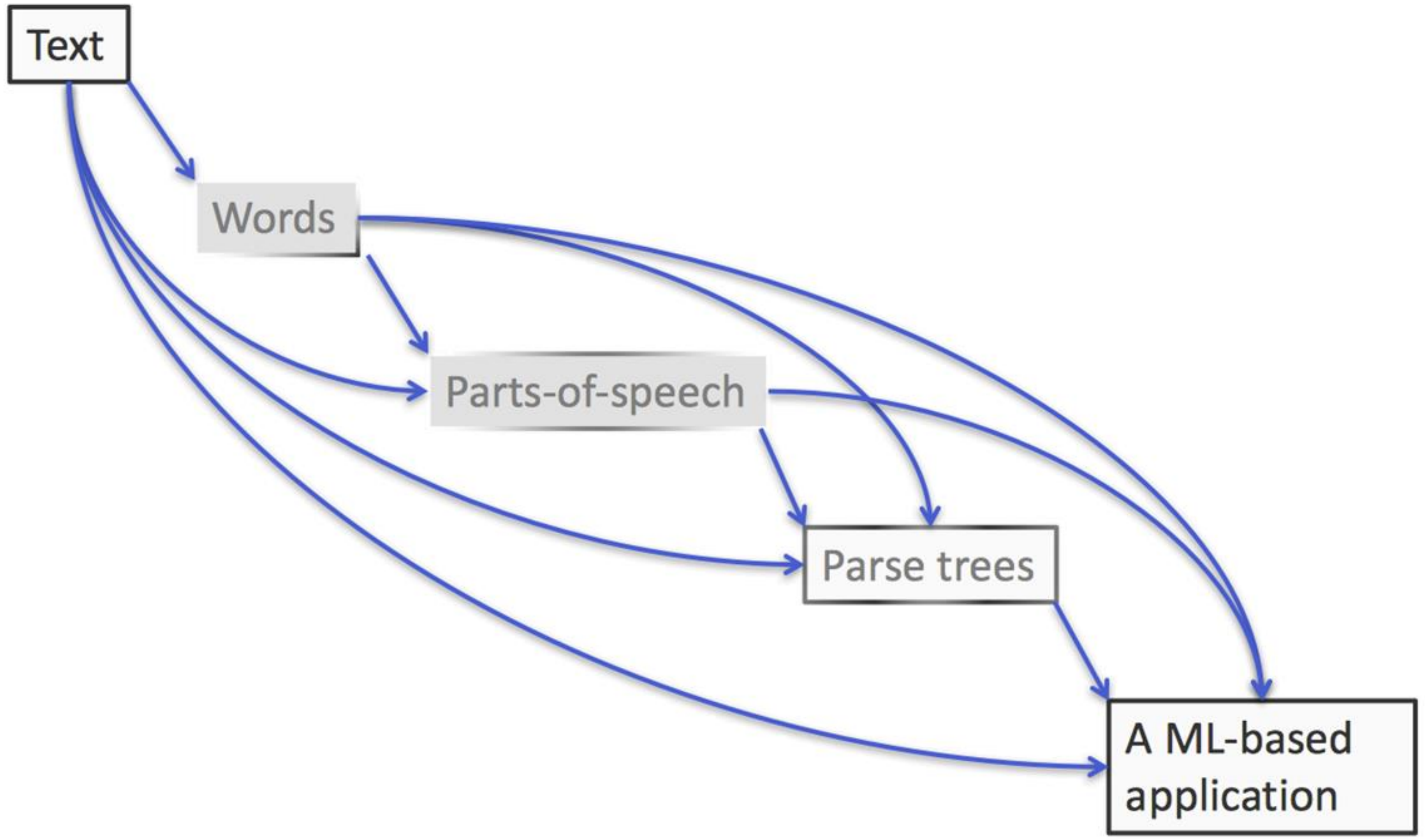

Each of these could be ML driven
Or deterministic but still **error prone**

How much do each of these contribute to the error of the final application?

# Example: text processing system

- Plug in the ground truth for the intermediate components and see how much the accuracy of the final system changes

| System | Accuracy |
|---|---|
| End-to-end predicted | 55% |
| With ground truth words | 60% |
| + ground truth parts-of-speech | 84% |
| + ground truth parse trees | 89% |
| + ground truth final output | 100% |



# Error analysis

- Explaining difference between the performance between a strong model and a much weaker one (a baseline)

- Usually seen with features
  - Suppose we have a collection of features and our system does well, but we don't know which features are giving us the performance
  - Evaluate simpler systems that progressively use fewer and fewer features to see which features give the highest boost

It is not enough to have a classifier that works; it is useful to know why it works

Helps interpret predictions, diagnose errors and can provide and audit trail

# Outline

- Model diagnostics
- Error analysis
- Additional advice

# Advice for ML workflow in practice

- Say you want to build a classifier that identifies whether a basketball player belongs to the **LA Lakers** or **Miami Heat**
- How do you go about this?

**The slow approach**

1. Carefully identify features, get the best data, the best software architecture, maybe design a new learning algorithm
2. Implement it and hope it works

**Advantage:** perhaps a better approach, maybe even a new learning algorithm. Research.

**The hacker's approach**

1. First implement something
2. Use diagnostics to iteratively make it better

**Advantage:** faster release, will have a solution for your problem quicker.

# Advice for ML workflow in practice

- Say you want to build a classifier that identifies whether a basketball player belongs to the **LA Lakers** or **Miami Heat**
- How do you go about this?

Be wary of premature optimization

Be equally wary of prematurely committing to a bad path

2. Implement it and hope it works

**Advantage:** perhaps a better approach, maybe even a new learning algorithm. Research.

**Advantage:** faster release, will have a solution for your problem quicker.

# What to watch out for?

- **Do you have the right evaluation metric?**
  - And does your loss function reflect it?

- **Beware of contamination:** ensure that your training data is not contaminated with the test set
  - Learning = generalization to new examples
  - Do not see your test set either. You may inadvertently contaminate the model
  - Beware of contaminating your features with the label!
  - Be suspicious of perfect predictors

# What to watch out for?

- Beware of bias vs. variance tradeoff (or overfitting vs. underfitting)

- Beware that intuitions may not work in high dimensions
  - No proof by picture
  - Curse of dimensionality

- A theoretical guarantee may only be theoretical
  - May make invalid assumptions (e.g. if the data is separable)
  - May only be legitimate with infinite data (e.g. estimating probabilities)
  - Experiments on real data are equally important

# What to watch out for?

- Learn simpler models first
- Ensembles seem to work better
- Think about whether your problem is learnable at all
  - Learning = generalization

# THANK YOU!